Chapter 20

Shortest Paths I: Spanners and Probabilistic Tree Embeddings

The topic of this chapter is *graph simplification*: How can we "simplify" a graph, say, reduce its edges and put more "structure" on it, while still preserving some of its main properties? In future chapters, we will see how this can play an important role in designing more efficient algorithms. For now, we consider a simple variant of this problem: how to simplify a graph while preserving its *shortest path structures* approximately.

20.1 Graph Spanners

Throughout this chapter, we will only work with *undirected* graphs. Given an undirected graph $G = (V, E)$, for any vertices $u, v \in V$, the distance between u and v—denoted by $\text{dist}_G(u, v)$—is the length of the shortest path between u and v. When the graph G is weighted, $\text{dist}_G(u, v)$ is the sum of the edge weights on the minimum-weight path between u and v, which we still call the shortest path. In this section, we focus on *unweighted* graphs.

Definition 20.1. For any (undirected) graph $G = (V, E)$, and any $\alpha \geqslant 1$, an α-**spanner** of G is any spanning subgraph H of G such that for all pairs of vertices $u, v \in V$:

$$\text{dist}_G(u, v) \leqslant \text{dist}_H(u, v) \leqslant \alpha \cdot \text{dist}_G(u, v).$$

(Note that the left inequality is redundant here given that H is a subgraph of G.)

It is easy to see that every graph admits an α-spanner for any $\alpha \geqslant 1$: simply let H be the same as G. Moreover, the only 1-spanner of any graph G is itself; even if we do not include a single edge (u, v) from G, then the distance between u and v will become more than 1 in the spanner, which violates the 1-spanner property. The trick however is to allow for some *approximation* of $\alpha > 1$ and obtain a *sparse* spanner, i.e., a one that has substantially fewer edges than G itself.

We can look at spanners from different points of view. An *algorithmic* approach for instance would be to find the *sparsest* α-spanner of a given graph G and parameter α. However, in this chapter (and generally in graph simplification questions), we are interested in an *extremal* question: can we prove that *every* n-vertex graph G admits an α-spanner with $O(f(n))$ edges for some function $f(n)$? We will answer this question by proving the following theorem.

Theorem 20.2. *For every integer $k \geqslant 1$, every graph G admits a $(2k-1)$-spanner with $O(n^{1+1/k})$ edges.*

Theorem 20.2 implies that there is already a 3-spanner for every graph with $O(n^{3/2})$ edges, much smaller than $O(n^2)$ edges, which is the total number of possible edges in G. Similarly, by setting $k = \log n$, we obtain that every graph G admits an $O(\log n)$-spanner with $O(n)$ edges.

It is worth asking why we are only interested in *odd* values of approximation factor α in α-spanners, i.e., $(2k-1)$-spanners in Theorem 20.2. The answer to this question hopefully will become clear from the proof of Theorem 20.2 but you are encouraged to think about it on your own also – in particular, what is the difference between a $2k$-spanner and a $(2k-1)$-spanner in a bipartite graph?

We prove Theorem 20.2 by presenting a simple algorithm, called the *greedy spanner*, for computing a $(2k-1)$-spanner of the input graph for any $k \geqslant 1$.

Algorithm 20.3.

1. Let $H \leftarrow \emptyset$ and $e_1, \ldots, e_m$ be any fixed ordering of the edges of G.
2. For $i = 1$ to m: if $H \cup \{e_i\}$ does not have a cycle of length $\leqslant 2k$, add e_i to H.
3. Return H as the spanner of G.

Let us first prove that H output by Algorithm 20.3 is a spanner of G.

Lemma 20.4. *The subgraph H of Algorithm 20.3 is a $(2k-1)$-spanner of the input graph G.*

Proof. We say that an edge $(u,v) \in G$ is **stretched** by α iff $\text{dist}_H(u,v) = \alpha$, i.e., in place of the edge (u,v) in H, we now have a path of length α. Any edge $(u,v) \in G \setminus H$ is stretched by at most $(2k-1)$ because the reason we did not add (u,v) to H was the existence of a cycle of length $\leqslant 2k$ in H (that inevitably includes both u and v); but this means that there is a path of length $\leqslant 2k-1$ between u and v in H. This proves the spanner property for every pair of vertices u, v that form an edge in G. We now extend this to all pairs. We do this using two equivalent but slightly differently written proofs. See Figure 20.1 for an illustration.

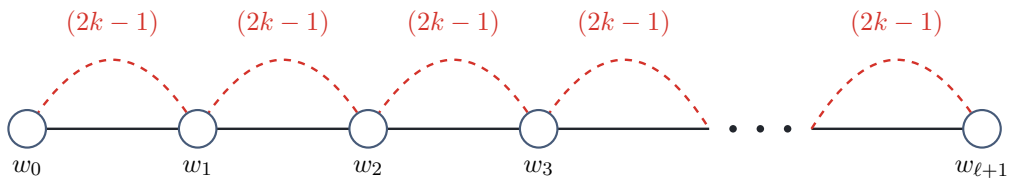

Figure 20.1: An illustration of the proof of Lemma 20.4. As we only skip edges of G in Algorithm 20.3 that are part of a $\leqslant 2k$ cycle, every edge of a shortest path between vertices x, y in G is stretched by at most $(2k-1)$ in H; thus, the length of x-y shortest path in H is at most $(2k-1)$ times larger than G.

Proof 1. Consider a pair of vertices $x, y \in V$ and the shortest path $x = w_0 \to w_1 \to w_2 \ldots w_{\ell-1} \to y = w_\ell$ in G with length ℓ. Every edge in this path is stretched by at most $(2k-1)$ in H and thus there is a *walk* of length $(2k-1) \cdot \ell$ in H between x, y. This immediately implies that $\text{dist}_H(x,y) \leqslant (2k-1) \cdot \text{dist}_G(x,y)$ as we can shortcut the edges of the walk into a path. Hence, H is a $(2k-1)$-spanner.

Proof 2. Again consider a pair of vertices $x, y \in V$ and the shortest path $x = w_0 \to w_1 \to w_2 \dots w_\ell \to y = w_{\ell+1}$ in G with length ℓ. We have,

$$\begin{aligned}\text{dist}_H(x,y) &\leqslant \sum_{i=1}^{\ell+1} \text{dist}_H(w_{i-1}, w_i) && \text{(by triangle inequality of distances)}\\ &\leqslant \sum_{i=1}^{\ell+1} (2k-1) \\ &\text{(as } (w_{i-1}, w_i) \text{ is an edge in } G \text{ and we have established the spanner property on edges)}\\ &= (2k-1) \cdot \text{dist}_G(x,y).\end{aligned}$$

This concludes the proof. □

Lemma 20.4 is kind of obvious by the design of Algorithm 20.3: we only skipped adding an edge to the spanner if we already provided a good approximation for it in the spanner. The main part of the analysis of the greedy spanner is then to bound its number of edges. Notice that a key property of the spanner H output by the algorithm is that it does not have any cycle of length at most $2k$ (the last edge of the cycle will never be inserted into the spanner). Now, we know that a graph without *any* cycle has to be sparse, i.e., can have at most $n-1$ edges. Can we generalize this to graphs that do not have "short" cycles?

20.1.1 Density of Graphs without Short Cycles

To do this, we rely on a classical result in graph theory referred to as the **Moore Bound**.

Proposition 20.5 (Moore Bound). *For any integer $k \geqslant 1$, any graph $G = (V, E)$ with no cycle of length less than or equal to $2k$ can have at most $n^{1+1/k} + n$ edges.*

Proof. Let m denote the number of edges in G. We first claim that G has an induced subgraph $\tilde{G}$ with minimum degree $\delta(\tilde{G}) > m/n$ (more than half the average degree). This can be proven as follows: if $\delta(G) > m/n$ we are done already by setting $\tilde{G} = G$; if not, pick any vertex $v \in G$ with $\deg(v) \leqslant m/n$ and remove it from the graph. As removing a vertex of degree at most m/n reduces the number of edges by at most m/n, this will reduce the average degree of the remaining graph to:

$$\geqslant \frac{2 \cdot (m - m/n)}{n-1} = \frac{2m \cdot (1-1/n)}{n \cdot (1-1/n)} = 2m/n.$$

As such, this process never decreases the average degree and hence needs to terminate in a non-empty subgraph $\tilde{G}$ with $\delta(\tilde{G}) > m/n$.

We now have a subgraph $\tilde{G}$ with minimum degree $d > m/n$ over at most n vertices. Consider growing a BFS tree of depth k from some arbitrary vertex v of $\tilde{G}$. The vertices of this tree cannot be visited in more than one path from v as otherwise any vertex x reachable from v via more than one path of length $\leqslant k$, creates a cycle of length $\leqslant 2k$ already, contradicting $\tilde{G} \subseteq G$ not having any $\leqslant 2k$-cycle (see Figure 20.2).

But this means that we should visit

$$1 + d + d \cdot (d-1) + d \cdot (d-1)^2 + \dots + d \cdot (d-1)^{k-1} \geqslant (d-1)^k$$

distinct vertices in this BFS tree. As the number of vertices of $\tilde{G}$ is at most n, we have,

$$(d-1)^k \leqslant n \implies d \leqslant n^{1/k} + 1 \implies \frac{m}{n} \leqslant n^{1/k} + 1 \implies m \leqslant n^{1+1/k} + n,$$

concluding the proof. □

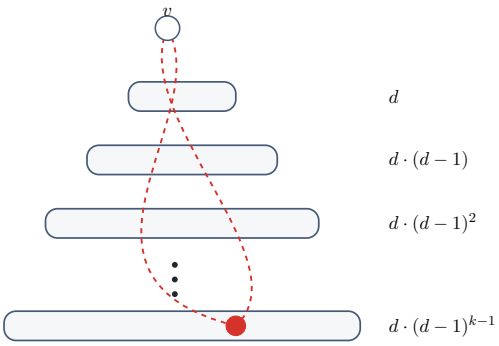

Figure 20.2: A simple illustration of the proof of Proposition 20.5. The vertices in the first k layers of the BFS tree from v are distinct, i.e., only have one path from v—otherwise, we will find a cycle of length $\leqslant 2k$ already (as shown by the red dashed edges).

Theorem 20.2 now follows immediately from Lemma 20.4 and Proposition 20.5.

The bound we obtained on the size of the spanner was a direct corollary of Proposition 20.5; thus, if we could improve the bounds there with a better analysis, we would immediately obtain a better bound on the size of the spanner as well. It is thus worth asking if we could improve Proposition 20.5? The answer turns out to be *No* in the following sense: for some small values of k, we already *know* this is impossible (to showcase this, we will prove this for $k = 2$ in the next section); for larger values of k, improving this bound or proving this is not possible is a longstanding open question in graph theory:

Remark. The **Erdős-Girth** conjecture states that:

> *For every integer $k \geqslant 1$, there are graphs with $\Omega(n^{1+1/k})$ edges with no cycle of length $\leqslant 2k$.*

In other words, the conjecture states that the Moore bound is asymptotically tight. This is an important conjecture when studying spanners (among others) but at this point, it is only known to be true for several small choices of k. The general conjecture is still open and the best known constructions imply a lower bound of $\Omega(n^{1+1/(2k-1)})$ edges (a random graph of average degree $\Theta(n^{1/(2k-1)})$ can be used to obtain this).

20.2 Asymptotic Optimality of Proposition 20.5 for $k = 2$

We now prove that there are graphs with no cycle of length $\leqslant 4$ which still have $\Omega(n^{3/2})$ edges. We first see a more "standard" approach using random graph theory (which we briefly saw in Chapter 3 that does *not* give the optimal bounds), and then see how to use a more clever (and "magical") construction that does the trick.

20.2.1 A "Weak" Construction Using Random Graphs

We prove the following weaker lemma.

Lemma 20.6. *For infinitely many $n \in \mathbb{N}$, there exists an n-vertex graph with no cycle of length at most 4 and $\Omega(n^{4/3})$ edges.*

Proof. Create a *bipartite* graph $G = (L, R, E)$ with $|L| = |R| = n$ by picking each edge inde-

pendently with probability $p \in (0,1)$ for some p to be determined later. Firstly, the expected number of edges in such a graph is:

$$\mathbb{E}\,|E| = p \cdot n^2.$$

Since G is bipartite, it cannot have a 3-cycle for us to worry about. What is a probability that this graph has a 4-cycle? We can calculate this probability by iterating over all possible choices of picking two vertices from both of L and R, and checking if all the edges between them appear in G. Thus,

$$\begin{aligned}
\Pr(G \text{ has a 4-cycle}) &\leqslant \sum_{u,v \in L, x,y \in R} \Pr(u,v,x,y \text{ form a cycle}) && \text{(by union bound)} \\
&= \sum_{u,v \in L, x,y \in R} p^4 && \text{(as all the 4 edges should appear independently)} \\
&= \binom{n}{2}^2 \cdot p^4 \\
&\qquad \text{(by the number of choices of two vertices from each of } L \text{ and } R) \\
&\leqslant n^4 \cdot p^4.
\end{aligned}$$

For this probability to be less than 1, we need to pick $p < 1/n$. But when $p < 1/n$, the expected number of edges in G is only going to be n, which is too small to be of any use.

The trick however is to use the *alteration method* discussed back in Section 12.2. Let C denote the number of 4-cycles in G. By the same exact calculation as above, we have that

$$\mathbb{E}\,[C] \leqslant p^4 \cdot n^4.$$

Now consider the random variable Y which is equal to the number of edges of G that do not appear in any 4-cycle. As each cycle counted in C contains 4 edges, we have $Y \geqslant |E| - 4 \cdot C$. Thus, by linearity of expectation,

$$\mathbb{E}\,[Y] \geqslant \mathbb{E}\,|E| - 4 \cdot \mathbb{E}\,[C] \geqslant p \cdot n^2 - 4 \cdot p^4 \cdot n^4.$$

Now, suppose we pick p such that

$$4 \cdot p^4 \cdot n^4 = \frac{1}{2} \cdot p \cdot n^2,$$

i.e., set

$$p := \frac{1}{2 \cdot n^{2/3}}.$$

Then, we have

$$\mathbb{E}\,[Y] \geqslant \frac{1}{2} \cdot p \cdot n^2 = \frac{1}{4} \cdot n^{4/3}.$$

This implies that there should exist a bipartite graph with $\Omega(n^{4/3})$ edges and no 4-cycles. Since bipartite graphs do not have a 3-cycle either, this concludes the proof. □

20.2.2 An Asymptotically Optimal Construction

We now provide a construction that proves the optimality of Proposition 20.5 for $k = 2$.

Lemma 20.7. *For infinitely many $n \in \mathbb{N}$, there exists an n-vertex graph with no cycle of length at most 4 and $\Omega(n^{3/2})$ edges.*

Proof. Let p be a prime number and define $\mathbb{F}_p$ as the field of integers modulo p, i.e., $\{0, 1, \ldots, p-1\}$, where addition, subtraction, and multiplication are done mod p.

Consider the following bipartite graph $G = (L, R, E)$ with the following vertices:

- Let $L := \mathbb{F}_p \times \mathbb{F}_p$ interpreted as *points* (x, y) in the two-dimensional plane ${\mathbb{F}_p}^2$.
- Let $R := \mathbb{F}_p \times \mathbb{F}_p$ interpreted as *lines* (a, b) in the two-dimensional plane ${\mathbb{F}_p}^2$.
- Let E be the set of edges that capture *incidence* in L and R meaning there is an edge between any $(x, y) \in L$ and $(a, b) \in R$, if the point (x, y) is on the line (a, b); formally, if

$$a \cdot x + b = y \pmod{p}.$$

See Figure 20.3 for an illustration.

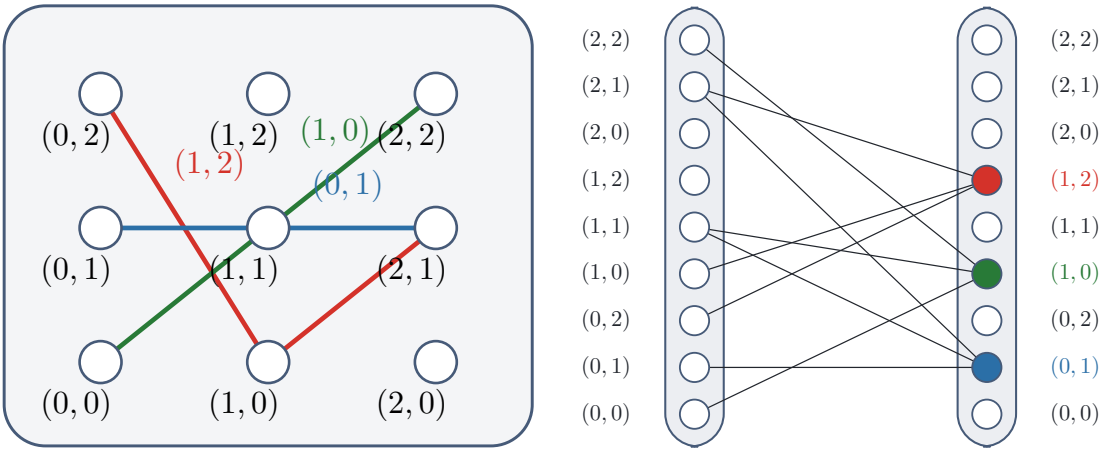

Figure 20.3: The plane used in the definition of the graph for $p = 3$ on the left, and the graph itself on the right. The points in $\{0, 1, 2\} \times \{0, 1, 2\}$ are all drawn – these correspond to vertices in L. Only three of the lines are drawn, which correspond to three marked vertices in R (the edges of remaining vertices in R are not drawn). Note that since the calculations are done modulo p, the lines (such as the red one) are not necessarily straight geometric lines.

We claim that this graph G satisfies all the properties we want.

Firstly, notice that the degree of every vertex in this graph is exactly p. Consider a vertex $(x, y) \in L$. For any $a \in \mathbb{F}_p$, there exists a unique $b = y - a \cdot x$, thus, (x, y) has p edges to vertices $(a, y - a \cdot x) \in R$ by ranging $a \in \mathbb{F}_p$. A similar analysis shows that degree of each $(a, b) \in R$ is also p (as each line in $\mathbb{F}_p \times \mathbb{F}_p$ contains p points inside it).

Secondly, G does not have any 4-cycles. Consider any pair of vertices (x_1, y_1) and (x_2, y_2) in L. Since there is at most one line (a, b) that passes through both these two points[1], (x_1, y_1) and (x_2, y_2) only have one shared neighbor. Thus, they cannot be part of a 4-cycle.

Finally, since $n = p^2$, we obtain a bipartite graph with $2n$ vertices and $n^{3/2}$ edges with no 4-cycles. Since bipartite graphs do not have a 3-cycle either, this concludes the proof. □

[1]Given that again our lines are defined mod p and not "geometrically", this may not be obvious at first glance. However, a simple linear algebra proves this: fixing both (x_1, y_1) and (x_2, y_2) gives us two linearly independent equations $a \cdot x_1 + b = y_1$ and $a \cdot x_2 + b = y_2$ for the two variables (a, b). As these two equations lead to a unique solution for (a, b), we can only have one line passing through both points.

20.3 Probabilistic Tree Embeddings

Theorem 20.2 shows that we can always reduce any given graph to a sparse subgraph with only $O(n)$ edges while preserving distances up to an $O(\log n)$ factor. But what if we require even more "structure" than just the sparsity of the resulting graph? In this section, our goal is to achieve a similar guarantee by reducing G to a *tree* (as in, a connected graph with no cycle). This way, not only do we achieve sharp bounds on the sparsity, but more importantly, we can now work with trees wherein shortest path computation is entirely trivial: there is only one path between each pair of vertices in a tree!

Before delving into the main result, let us first consider some naive attempts on approximating distances of a given graph with graphs with simpler structures. Given a graph G, consider "simplifying" G by a complete graph K such that for all $u, v \in V$, $w_K(u,v) = \text{dist}_G(u,v)$, where $w_K(u,v)$ is the weight of the edge (u,v) in K. Then, to find the shortest path of any u, v, we simply look up the weight of this edge in K. This preserves the distances in G *exactly*, but can be very time consuming to construct, and takes up a lot of space, so it is not clear how much "simpler" it is (although computing the distances from it is indeed simple).

Another approach would be to embed a graph G into a *single* tree such that $\text{dist}(u,v)$ is approximately preserved for all $u, v \in V$. A simple sanity check shows that such guarantee cannot always exist. Consider a cycle graph of n vertices: any way of turning the graph into a tree must remove at least one edge (u,v), which changes $\text{dist}_G(u,v)$ from 1 to $n-1$, thus, nowhere close to a good approximation of $\text{dist}_G(u,v)$.

A workaround to the issue above—that a single tree can never preserve all distances in many graphs—was first introduced by a beautiful result of [Bar96], referred to as *probabilistic tree embedding*. The general goal is to consider a distribution of trees $\mathcal{D}$ such that the distance between any pair of vertices is approximately preserved in expectation, i.e. $\forall u, v \in V$,

$$\mathop{\mathbb{E}}_{T \sim \mathcal{D}}[dist_T(u,v)] \approx \text{dist}_G(u,v);$$

i.e., if we fix a pair of vertices and then sample a tree from this distribution, the vertices will have roughly the same distance in T and G in expectation. We should emphasize that here $V(T) \supseteq V(G)$, i.e., we allow additional vertices to appear in T as well.

Theorem 20.8 ([Bar96])**.** *For any undirected graph $G = (V, E)$, there exists a distribution $\mathcal{D}$ of weighted trees $T_1, \cdots, T_k$ such that for all $i \in [k]$, we have $V(G) \subseteq V(T_i)$ and that for any fixed pair of vertices $u, v \in V(G)$,*

$$\text{dist}_G(u,v) \leqslant \mathop{\mathbb{E}}_{T \sim \mathcal{D}}[\text{dist}_T(u,v)] \leqslant O(\log^2 n) \cdot \text{dist}_G(u,v).$$

We call the $O(\log^2 n)$ factor above the **stretch** *of the tree embedding.*

This result relies on another important topic in algorithm design: *Low Diameter Decomposition (LDD)*. We first review this topic and then show how to use it to prove Theorem 20.8.

20.4 Low Diameter Decomposition (LDD)

We start with some definitions.

Definition 20.9. Given a graph $G = (V, E)$, the **diameter** of G is defined as

$$D(G) := \max_{u,v \in V} \text{dist}_G(u,v),$$

namely, the distance between the "furthest" pairs of vertices. For any subset $S \subseteq V$, the **weak diameter** of S is defined as

$$D(S) := \max_{u,v \in S} \text{dist}_G(u, v).$$

Note that the weak diameter of S is *not* the diameter of the induced subgraph $G[S]$, as we still calculate the distances in the entire graph G and not just $G[S]$; this is the reason this is called the *weak* diameter. Finally, for any integer $r \geqslant 1$ and any vertex $v \in V$, we define

$$Ball(v, r) := \{u \in V \mid \text{dist}_G(u, v) \leqslant r\}$$

and refer to it as the **ball of radius** r centered at v. We can now define LDDs as follows.

Theorem 20.10. *Given any undirected graph $G = (V, E)$ and parameter $D \geqslant 0$, there exists a random partition $V = C_1 \sqcup C_2 \sqcup \cdots \sqcup C_k$ such that:*

1. *$D(C_i) \leqslant D$ for all $i \in [k]$, i.e., weak diameter of each "cluster" C_i is at most D (deterministically);*
2. *For all $u, v \in V$,*
$$\Pr\left(C(u) \neq C(v)\right) = \frac{O(\log n)}{D} \cdot \text{dist}_G(u, v),$$
where $C(u)$ denotes the partition containing vertex u.

This theorem states that we can always partition a graph into disjoint components such that vertices inside each component have short distances with each other and vertices that are close in the original graphs are unlikely to be assigned to different components.

We show the following procedure produces a partition $C_1 \sqcup \cdots \sqcup C_k$ of V that satisfies the two properties in LDD with high probability.

Algorithm 20.11. Start with all vertices in V being unmarked. Repeat while there is an unmarked vertex:

1. Pick any arbitrary unmarked vertex v.
2. Sample R_v from the geometric distribution of success probability $p := \dfrac{10 \log n}{D}$ on each trial[a].
3. Put all *unmarked vertices* in $B(v, R_v)$ into a cluster. Mark all of them.

[a] That is R_v is distributed as the number of trials we need to run before getting the first success.

Let us first show that the weak diameter of each cluster will be at most D with high probability.

Claim 20.12. *For any cluster C output by Algorithm 20.11, $D(C) \leqslant D$ with high probability.*

Proof. Let v be the center of the cluster C, namely, the vertex picked in Algorithm 20.11 that defined C as the intersection of $B(v, R_v)$ with unmarked vertices. Any pair of vertices u, w in C have

$$\text{dist}_G(u, w) \leqslant \text{dist}_G(u, v) + \text{dist}_G(v, w) \leqslant R_v + R_v,$$

by triangle inequality and since both $u, w \in B(v, R_v)$. We argue that with high probability, $R_v \leqslant D/2$ which happens because

$$\Pr\left(R_v > \frac{D}{2}\right) \leqslant (1-p)^{D/2} \leqslant \exp\left(-\frac{10\log n}{D} \cdot \frac{D}{2}\right) \leqslant n^{-5};$$

the rest follows from a union bound over the at most n clusters. □

We now prove the second property. The main part is to prove it for adjacent vertices, which is done by the following claim – we will then show how this easily implies the property for all pairs of vertices.

Claim 20.13. *For any edge (u, v),*

$$\Pr\left(C(u) \neq C(v)\right) = \frac{O(\log n)}{D}.$$

Proof. Let $(u, v) \in E$. We have

$$\begin{aligned}\Pr\left(C(u) \neq C(v)\right) = \Pr\left(u \text{ was clustered first}\right) \cdot \Pr\left(C(u) \neq C(v) \mid u \text{ was clustered first}\right) \\ + \Pr\left(v \text{ was clustered first}\right) \cdot \Pr\left(C(u) \neq C(v) \mid v \text{ was clustered first}\right).\end{aligned}$$

Let us consider the term

$$\Pr\left(C(u) \neq C(v) \mid u \text{ was clustered first}\right).$$

Let w be the vertex which clustered u. By conditioning on u being clustered first, we have that

$$R_w \geqslant \text{dist}_G(w, u).$$

At the same time, for $C(u) \neq C(v)$ to happen, we should have

$$R_w < \text{dist}_G(w, v) \leqslant \text{dist}_G(w, u) + 1,$$

where the second inequality is because there is an edge (u, v) in the graph. Thus, we have,

$$\begin{aligned}\Pr\left(C(u) \neq C(v) \mid u \text{ was clustered first}\right) &= \Pr\left(R_w < \text{dist}_G(w, u) + 1 \mid R_w \geqslant \text{dist}_G(w, u)\right) \\ &= p = \frac{O(\log n)}{D}.\end{aligned}$$

Since we can bound the term when v is clustered first exactly the same way also, we can conclude the proof of the claim. □

Finally, we extend the property of Claim 20.13 to all pairs of vertices (u, v). Consider a shortest path $P_{uv} = w_1, \ldots, w_{d+1}$ between u and v where $w_1 = u$, $w_{d+1} = v$ and $d = \text{dist}_G(u, v)$. By union bound,

$$\begin{aligned}\Pr\left(C(u) \neq C(v)\right) &= \Pr\left(\bigvee_{i=1}^{d} C(w_i) \neq C(w_{i+1})\right) \\ &\leqslant \sum_{i=1}^{d} \Pr\left(C(w_i) \neq C(w_{i+1})\right) \leqslant \text{dist}_G(u, v) \cdot \frac{O(\log n)}{D},\end{aligned}$$

by applying Claim 20.13 to each edge (w_i, w_{i+1}).

Before concluding the proof of Theorem 20.10, we should note that technically, we only showed $D(C_i) \leqslant D$ holds with high probability. To make this a deterministic statement, we can modify Algorithm 20.11 such that when there is a cluster that violates the diameter constraint, we output the trivial output where every vertex in that cluster is now its own cluster. Since the probability of this occurring is at most $1/n^4$ by Claim 20.12, the increase in the error probability of $\Pr(C(u) \neq C(v))$ is negligible and the previous bounds still hold asymptotically. But now we have a deterministic upper bound on the weak diameter of each cluster, and can conclude the proof of Theorem 20.10.

20.5 Back to Tree Embeddings: Proof of Theorem 20.8

We now go back to probabilistic tree embeddings and prove Theorem 20.8. We construct a tree embedding of G by recursively finding LDDs in G for geometrically decreasing values of the weak diameter. The algorithm is given a graph $G = (V, E)$, a set $U \subseteq V$ of vertices to be clustered, and a parameter D as an upper bound on the weak diameter of U in G. To obtain a tree embedding of a given graph G, we run this algorithm for (G, V, n).

Algorithm 20.14.

1. If U only has a single vertex, return that single vertex as the tree embedding.
2. Let $C_1 \sqcup \cdots \sqcup C_k$ be an LDD of U in G with parameter $D/2$ using Theorem 20.10.
3. For each $i \in [k]$, recursively compute a tree embedding of $(G, C_i, D/2)$ called T_i.
4. Create a new tree T using a new root r which is connected to the roots of each of the sub-trees T_i's for $i \in [k]$, using an edge of weight $D/2$. Return T as the tree embedding of (G, U, D).

First observe that the vertices in G are the leaves in the output tree T. Roughly speaking, the procedure recursively splits vertices into different groups at each level based on their distance. For any $u, v \in V$, if $\text{dist}_G(u, v)$ is small, then they are likely to be split towards the end of the algorithm, hence the smallest subtree in T that contains u, v is expected to be small and subsequently $\text{dist}_T(u, v)$ is small. If $\text{dist}_G(u, v)$ is large, then they are expected to be separated early on, hence $\text{dist}_T(u, v)$ is large.

By construction, for all possible tree T output by the algorithm we have $V(G) \subseteq V(T)$, which gives the first property of the embedding.

For the second property, we first claim that $\text{dist}_T(u, v) \geqslant \text{dist}_G(u, v)$ for all possible output tree T. Consider the recursive call that separates u and v and let D be the weak diameter parameter of that call. Since u and v are not separated yet, we have $\text{dist}_G(u, v) \leqslant D$ since weak diameter of a recursive call with value D is at most D. On the other hand, since u and v are now separated, we have, $\text{dist}_T(u, v) \geqslant D$ as we need to traverse two edges of weight $D/2$ to go from u to the root of T and then to v, hence, we always have

$$\text{dist}_T(u, v) \geqslant \text{dist}_G(u, v),$$

deterministically (and thus certainly in expectation).

Finally, we argue that expected stretch of T cannot be too large either. We have

$$\mathop{\mathbb{E}}_{T}[\text{dist}_T(u,v)] \leqslant \sum_{i=0}^{\log(n/\text{dist}_G(u,v))} \Pr\big(u,v \text{ separated in a call } D \text{ with } 2^i \cdot dist_G(u,v) \leqslant D < 2^{i+1} \cdot \text{dist}_G(u,v)\big) \cdot 2D$$

(by the construction of the algorithm)

$$= \sum_{i=0}^{\log(n/\text{dist}_G(u,v))} \underbrace{\frac{O(\log n)}{2^i \cdot \text{dist}_G(u,v)} \cdot \text{dist}_G(u,v)}_{\text{by Theorem 20.10}} \cdot 2 \cdot \underbrace{\big(2^{i+1} \cdot \text{dist}_G(u,v)\big)}_{\text{upper bound on } D}$$

$$= O(\log^2 n) \cdot \text{dist}_G(u,v).$$

This concludes the proof of Theorem 20.8 and our study of probabilistic tree embeddings.

Remark. The stretch of tree embeddings in Theorem 20.8 by [Bar96] was subsequently improved to $O(\log n)$ in [FRT03] which is the *optimal* bound for this problem.